SPECIFICATION

Please amend the application as follows:

Please add the following paragraph at page 1, after the title:

This application is a continuation of U.S. Patent Application Serial No. 09/475,417, filed December 30, 1999, which is incorporated herein by reference.

The paragraph beginning at page 2, line 21 is amended as follows:

Multiple processor (MP) computer systems further ~~complicates~~ complicate the problems of error recovery and error recovery time. In MP computer systems, different processors are executing different processes. One or more of the processors may encounter the error but all of the processors can be affected. Generally, MP computer systems lack a coordinated approach to error recovery. This lack of an appropriate error handling can cause MP computer systems to reboot unnecessarily and data to be corrupted.

The paragraph beginning at page 4, line 24 is amended as follows:

Firmware as used herein refers to processor routines that are stored in non-volatile memory structures such as read only memories (ROMs), flash memories, and the like. These memory structures preserve the code stored in them even when power is shut off. Even though firmware is stored in non-volatile memory, firmware may be copied or shadowed to volatile memory. Typically, this is done for performance reasons. One of the ~~principle~~ principal uses of traditional firmware is to provide necessary instructions or routines that control a computer system when it is powered up from a shut down state, before volatile memory structures have been tested and configured. Firmware routines may also be used to reinitialize or reconfigure the computer system following various hardware events and to handle certain platform events like system interrupts.

The paragraph beginning at page 6, line 1 is amended as follows:

For a single processor system, there are only local MCAs. The MCAs in the single processor computer system are not divided between global and local since there are not other processors in the computer system.

The paragraph beginning at page 8, line 5 is amended as follows:

PAL 201 includes a set of procedures and an error handling routine. The set of procedures ~~for accessing~~ access processor hardware, and can encapsulate processor model specific hardware. This set of procedures may be called by other system software or hardware. By providing the set of procedures, PAL 201 provides a consistent interface to access processor resources across different processor implementations. The error handling routine of PAL 201 is also known as PAL_CHECK. Error handling for an error may be handed off to the error handling routine of PAL 201 or PAL_CHECK.

The paragraph beginning at page 12, line 3 is amended as follows:

SAL 302 is a platform specific firmware component that is typically provided by original equipment manufacturers (OEM) and BIOS vendors. The SAL is a firmware layer that isolates an operating system and other higher level software from implementation differences in the platform. There is a PAL-SAL software interface that allows communication between these PAL and SAL layers. The PAL-SAL interface is of a predetermined and standardized format. SAL 302 includes an error handling routine that, when executed by a processor, performs error handling. The SAL ~~303~~ error handling routine is branched to once the PAL error handling routine has finished processing the error or attempting to correct the error. The SAL error handling routine checks the status of the error. If the error is uncorrected, the SAL error handling routine determines the severity of the error, creates a log regarding the error or adds information regarding the error to a log and attempts to correct the error. The severity of the error is determined by using the PAL error information. The SAL error handling routine may use PAL procedures or PAL APIs to gain additional information about the error in order to determine the

severity of the error. If the error is corrected, the SAL error handling routine creates a log regarding the error or adds information regarding the error to an existing log.

The paragraph beginning at page 13, line 1 is amended as follows:

SAL 303 includes a GET_INFO procedure or API for getting state information. The procedure provides a programmatic interface to processor and platform information logged by SAL 302 with respect to the machine state at the time of errors including MCAs and CMCs. The procedure may be called by the OS ~~304~~ 303 or any other diagnostic software. The procedure provides a standardized format for accessing logged information. The amount of state information saved by SAL is implementation dependent. SAL 302 can provide an indication of what information SAL is logging. This may be done, for example, by providing validation bits indicating the saved state information. The procedure may clear the log if instructed to do so by the calling OS 303 or other diagnostic software. The log may be stored in a register. The log is generally cleared on booting the system. In the case of multiple errors of the same type, the log will provide details and information relating to the first occurring error. The call may specify whether it is requesting a platform log or processor log. The procedure can write the log information to a region of memory specified by the calling software.